ROBOT VISION

Vision:

Robot vision is the important part in robotics which refers to capability of perceiving the environment, using this information for execution of different tasks like pick and place, autonomous navigation, object recognition etc. Vision is implemented in order to give the visual feedback to the robot system to achieve the tasks at greater accuracy.

Vision sensor required for Vision:

A **Kinect** v3 sensor by Microsoft is used to get the visual data. Kinect consists of,

1. Color VGA sensor.
2. IR projector and receiver.
3. Microphone array as shown below.

Fig: KINECT CAMERA SENSOR.

Why kinect is used?

In earlier versions a stereo camera is used for vision, which is gives an image in gray scale so object recognition was difficult. But kinect provides a RGB image through which object recognition is made easy and it provides depth image of the environment which helps in depth calculation of the objects.

Working of Kinect:

With the help of IR projector the kinect transmits an IR rays in specific pattern called structured light these rays get reflected when they hit the obstacle these reflected rays is sensed by the IR receiver and from this data with the help of kinect depth perception it able to estimate the depth in distance of each every pixel in its frame. To calculate the distance it uses ToF (Time of Flight) or structured triangulation in Kinect v3 it uses ToF.

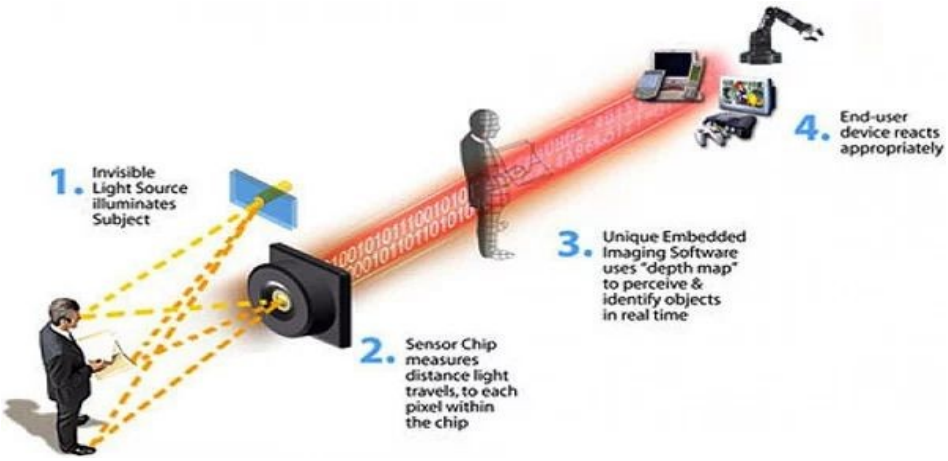

Fig: Describing basic working of KINECT

Disadvantages of Kinect usage:

Because of its weight and dimensions it is very difficult to handle, since it's like eye to human it is be placed in head of the robot so it is very difficult to move up-down of head and it requires very high torque servo motors.

Object Recognition and Detection:

The data from color sensor of kinect which we are receiving is of only RGB image in jpeg format, this image is used for object recognition and detection. Object detection is done using CNN (convolution neural networks) i.e. is through deep learning concepts. There are different object detection algorithms,

1. Single shot detection (SSD)
2. RetinaNet
3. You Only Look Once (YOLO)

In many visions yolo object detection method is used because of its higher accuracy than compared to others, yolo works on neural network with darknet53 architecture and it able detect around 80 classes.

Working of YOLO:

The input to the yolo network is rgb image, this image is down-sampled by a factor called stride of the network (example: 416x416 is passed with a stride of 32 the output size is 13x13) and the resulting output is called as feature map this is passed to the network with weights learned on coco dataset using ImageNet, the convolution layer in yolo is 1x1 so the output of the layers remains same as that of the feature map, feature map is made of grid cells (13x13 grid cells with stride factor 32 and input image 416x416) here each cell is used for the prediction of object, but the cell containing centre of the ground truth box of an object is responsible for the prediction of the object as shown in below.

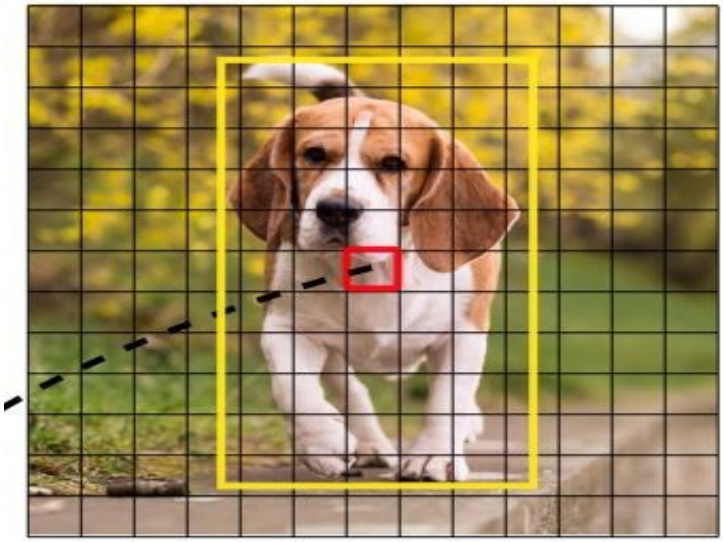

Fig: Image representing grid cells, Bounding Box and the grid cell (red cell) responsible for the object detection and recognition.

This cell now predicts three bounding boxes among that, one is selected which has higher IoU (Intersection over Union w.r.t ground-truth bounding box) this is known as NMS(Non maximum suppression) there may be a chances that multiple cells detect same object, so in this case also NMS is used to predict bounding box the below shows how NMS works.

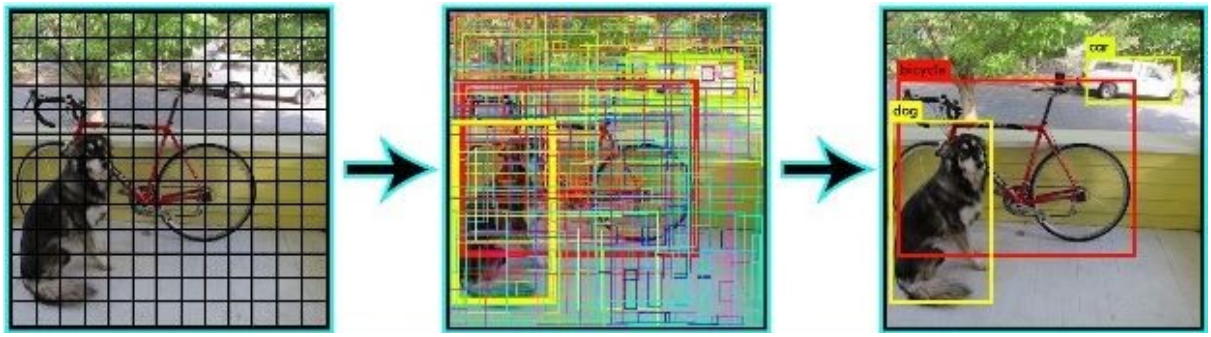

Fig: Representing how prediction occurs and NMS is implemented in selecting the bounding box.

Each bounding box has many attributes like box co-ordinates, object score and class scores as shown in image.

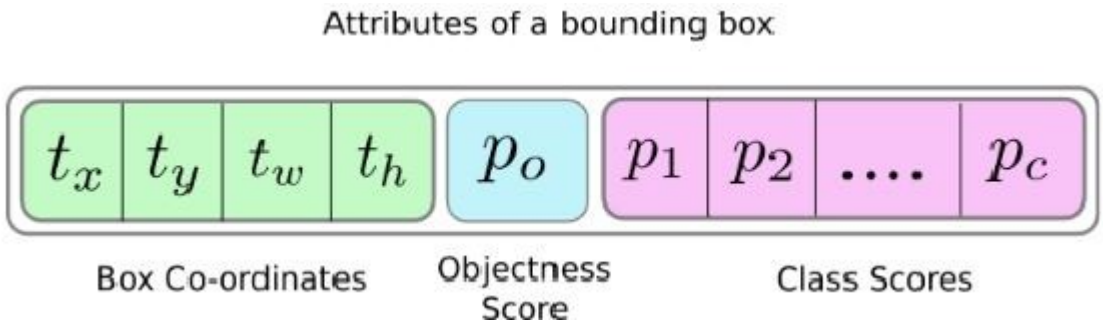

Fig: Array containing the attributes the bounding box which is an result of neural network.

Box co-ordinates represents the dimension w.r.t to the grid cell this has to be transformed to the original image dimension as shown below.

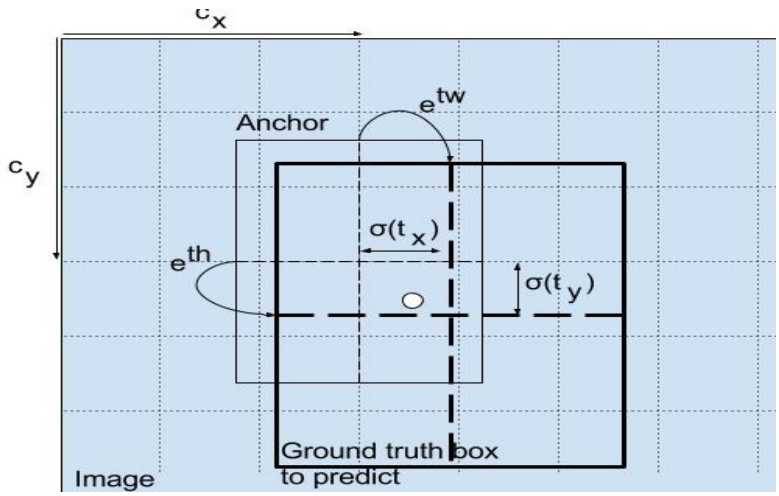

Fig: Transform from coordinates predicted grid cell to coordinates of input image.

Here, tx ty tw th are the output of the network which are co ordinates of the centre point of the grid cell which is responsible for the prediction this has to be transformed to input image bounding box co-ordinates.

Object score gives the probability of an object contained in the box and finally class scores/confidence gives the probability of detected object w.r.t all the classes (here in coco dataset, C = 80).

One of the main feature in yolo is, it's possible to predict both bigger and smaller objects relative to the image, the technique used here is Prediction across different scales yolo uses 3 scale with different stride factor(32 at 1st scale, 16 at 2nd scale and 8 at 3rd scale) as we seen stride factor is responsible for cells creation it decides the size of the cell and number of the cells in an feature map therefore number cells increases with decrease in stride factor as shown in the figure below. So in order to detect both low and high level features different stride is used at different scales. And the results at each scale are concatenated.

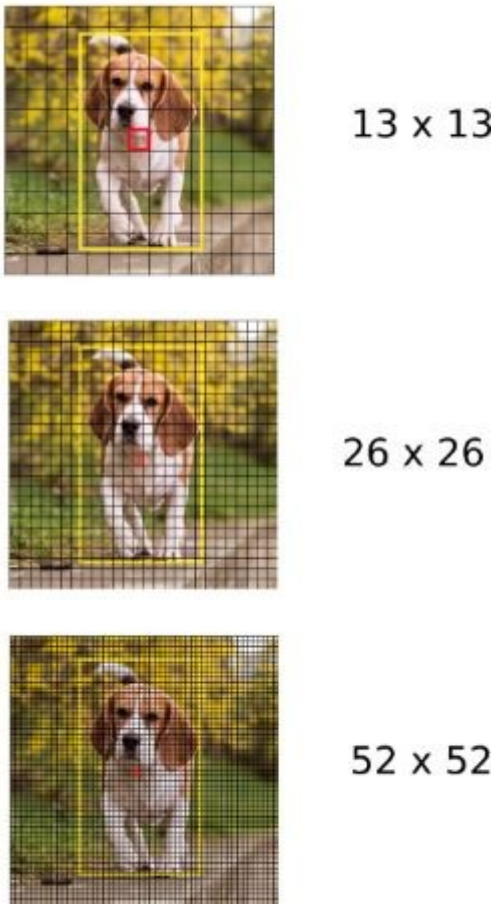

Fig: Image subjected to different stride factor to get the feature map.

Depth data from kinect:

The kinect depth raw image stores the depth value of each and every pixel, using this depth value of required pixel can be obtained and this gives us how far the object w.r.t camera centre. And finally the real world co-ordinates (x, y, z) of the detected object w.r.t camera is obtained which is used for robotic applications.